INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-24 06:18 UTC

AI Agent Skill Marketplaces Structurally Vulnerable to Post-Scan Payload Substitution

SECURITY ANALYSIS | **HIGH** | CVSS 7.5

SCC Item ID	SCC-STY-2026-0261
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	AI agent skill marketplaces broadly; Cisco skill scanner, NVIDIA skill scanner, skills.sh scanners, ClawHub (Trail of Bits research), Google Stitch (impersonated); AI agents consuming third-party skills
Published	2026-06-23T11:16:43
Discovery Source	Rss

Executive Summary

Security researchers at AIR demonstrated a structural bypass affecting AI agent skill marketplaces from multiple vendors: a skill package submitted for scanning can contain no malicious content at scan time, then silently swap in a malicious payload at runtime by rewriting an external URL after approval. A single test skill reached approximately 26,000 agents, including corporate deployments. This is not a one-vendor bug, it is a design flaw baked into the one-time-scan model that underpins the entire skill marketplace ecosystem, and independent research from Trail of Bits confirms the problem is widespread. (No CVE assigned as of publication date; this is a structural design flaw rather than a discrete vulnerability.)

Technical Analysis

The attack chain exploits a fundamental mismatch between how skills are scanned and how they execute. A threat actor submits a skill package that is genuinely clean at submission time, no malicious code, no suspicious strings, nothing for a static analyzer to flag. The package passes vendor scanning (Cisco, NVIDIA, and others tested by AIR). Once approved and distributed, the skill fetches a dependency or payload from an external URL at runtime. The attacker then rewrites that URL's content to deliver malicious code. The scan result is permanently stale from the moment the external resource changes. This maps directly to CWE-494 (Download of Code Without Integrity Check) and CWE-829 (Inclusion of Functionality from Untrusted Control Sphere), the skill scanner has no visibility into what the runtime-fetched resource becomes after approval. CWE-345 (Insufficient Verification of Data Authenticity) and CWE-346 (Origin Validation Error) also apply, as the marketplace does not verify that externally fetched content originates from the expected source or has not been

modified in transit.

The supply chain dimension compounds the problem. Trail of Bits documented related structural weaknesses in skill distribution infrastructure in their June 2026 ClawHub research, independently corroborating AIR's findings from a different angle. A separate analysis found that seven skill scanners agree on only 0.12% of detections, an extraordinary inconsistency that signals the scanner ecosystem lacks any shared baseline for what constitutes malicious behavior in a skill package. Attackers do not need to defeat a strong scanner; they need to defeat any one scanner in a fragmented field where consensus barely exists.

The MITRE ATT&CK mapping illuminates the full kill chain: T1195.002 (Compromise Software Supply Chain) and T1195.001 (Compromise Software Dependencies and Development Tools) capture the initial distribution vector; T1608.001 (Stage Capabilities: Upload Malware) describes the post-approval payload staging; T1102 (Web Service) and T1071.001 (Application Layer Protocol: Web Protocols) cover the runtime callback mechanism; T1059 (Command and Scripting Interpreter) addresses downstream execution once the payload lands. The technique also incorporates T1036.005 (Masquerading: Match Legitimate Name or Location) and T1027.012 (Obfuscated Files or Information: LNK Icon Smuggling) to maintain the appearance of a legitimate skill.

The claim that approximately 26,000 agents, including corporate accounts, received the test skill before any intervention underscores the blast radius problem. Unlike traditional software supply chain attacks that require compromising a trusted build pipeline, this technique requires only marketplace approval of a clean package, a controllable external URL, and patience. The barrier to entry is low; the distribution multiplier is high.

Action Checklist

1. Step 1: Assess exposure, audit every AI agent deployment in your environment and catalog all third-party skills currently installed; identify which marketplace or distribution channel each skill originated from, specifically including Cisco AI Defense, NVIDIA, skills.sh (a public AI agent skill distribution platform), and any compatible skill marketplaces.
2. Step 2: Review controls, verify whether your AI agent runtime environment enforces integrity checks on externally fetched dependencies (CWE-494 / CWE-829 mitigations); check whether network egress controls (NIST AC-4, Information Flow Enforcement) block or inspect outbound skill runtime callbacks to external URLs; confirm whether skills execute in isolated sandboxes with NIST AC-6 (Least Privilege) applied to the agent process.
3. Step 3: Inventory runtime fetch behavior, for each installed skill, determine whether it makes outbound HTTP/S calls at execution time; document the external URLs or domains contacted, and verify those resources are controlled by the original publisher, not rewriteable by a third party.
4. Step 4: Apply continuous integrity verification, a one-time scan at submission is insufficient; implement or demand from your marketplace vendor a mechanism that re-verifies external dependency content at each execution (content hashing, signed manifests, or pinned artifact digests); align with D3FEND countermeasures such as D3-PBWSAM (Proxy-based Web Server Access Mediation) and D3-LAM (Local Account Monitoring) extended to runtime-fetched content.
5. Step 5: Restrict skill installation permissions, apply NIST AC-2 (Account Management) and CIS 6.1 (Establish an Access Granting Process) principles to skill installation: require change-control approval, limit who can add or update agent skills to dedicated administrator accounts (CIS 5.4), and log all skill install and update events per NIST AU-2 (Event Logging).

6. Step 6: Update threat model, add T1195.002 (Supply Chain Compromise) and T1102 (Web Service runtime callback) as active TTP patterns in your AI agent threat register; document the post-scan payload substitution pattern as a distinct attack scenario separate from traditional software supply chain compromise.

7. Step 7: Communicate findings, brief leadership that the risk is not a patched vulnerability but an architectural gap in the marketplace model itself; frame it as a vendor assurance and procurement control issue requiring policy changes, not a one-time remediation.

8. Step 8: Monitor developments, track Trail of Bits ClawHub disclosures, Cisco AI Defense skill-scanner updates, and NVIDIA security advisories for any scanner improvements or structural mitigations; watch for CISA or NIST AI Risk Management Framework (AI RMF) guidance addressing skill marketplace integrity.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal counsel if DNS log analysis (Step 3) reveals that any installed skill's runtime URL resolved to an IP or domain different from its state at marketplace approval time, indicating live post-scan substitution may have already executed against production agents — particularly if those agents had access to systems processing PII, PHI, or financial data triggering breach notification obligations.
Recovery Notes	Before returning any quarantined AI agent deployment to production, verify that every installed skill's external dependency hashes match the approved pinfile baselines established in Step 4, and confirm via Wireshark or tcpdump that no unauthorized outbound runtime callbacks occur during a controlled test execution. Maintain heightened DNS and network egress monitoring on agent hosts for a minimum of 30 days post-remediation, as the post-scan substitution mechanism can be reactivated at any time by an adversary who still controls the external URL without any new skill installation event occurring.
Forensic Artifacts	Skill manifest files (YAML/JSON) from the agent skill installation directory with SHA-256 hashes and file modification timestamps — delta between hash at marketplace approval and hash at runtime is the primary indicator of post-scan substitution | DNS resolution logs (Windows DNS Client Event Log Event ID 3008; Linux /var/log/syslog or auditd) showing which IP addresses skill-associated external domains resolved to at each execution, enabling timeline reconstruction of when a URL was rewritten | Full PCAP from skill execution network traffic captures (tcpdump/Wireshark) showing HTTP/S request headers, response bodies, and TLS SNI values for all outbound runtime callbacks made by the skill process | HTTP response headers (Last-Modified, ETag, Content-Length, X-Cache) retrieved via curl -I from each skill's declared external dependency URL — changes in these values relative to the marketplace scan date indicate post-approval content substitution | Audit log entries for skill installation directory writes (Linux auditd key 'skill_install'; Windows Security Event ID 4663 Object Access) and agent process network connections (Sysmon Event ID 3) correlated against change-control approval records to identify unauthorized skill installs or updates

Per-Action IR Details

Step 1: Assess exposure — audit every AI agent deployment in your environment and catalog all third-party skills currently installed; identify which marketplace or distribution channel each skill originated from, specifically including Cisco AI Defense, NVIDIA, skills.sh, and any OpenClaw-compatible marketplace.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: scope and impact assessment of potentially affected systems

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Export installed skill manifests using each platform's CLI or API (e.g., `openai tool list`, Cisco AI Defense inventory export, or NVIDIA NIM skill registry query). Cross-reference against a manually maintained spreadsheet tracking skill name, version hash, source marketplace, and declared external URL dependencies. A 2-person team can automate this with a simple bash or Python script that iterates the agent config directory and extracts all `url:` or `endpoint:` fields from skill YAML/JSON manifests.

Evidence: Before any action that alters agent configuration: snapshot the current skill manifest files (e.g., `~/.agent/skills/*.yaml`, platform-specific config directories), capture their SHA-256 hashes (`sha256sum`), and record the currently resolved IP addresses of all external URLs declared in those manifests using `curl -I` or `dig` with a timestamp. This establishes a forensic baseline of what the skill pointed to at audit time versus what it pointed to at scan/approval time — the core artifact for detecting post-scan URL substitution.

Step 2: Review controls — verify whether your AI agent runtime environment enforces integrity checks on externally fetched dependencies (CWE-494 / CWE-829 mitigations); check whether network egress controls (NIST AC-4, Information Flow Enforcement) block or inspect outbound skill runtime callbacks to external URLs; confirm whether skills execute in isolated sandboxes with NIST AC-6 (Least Privilege) applied to the agent process.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: characterizing the attack surface and existing control gaps relevant to the observed threat pattern

Controls: NIST AC-4 (Information Flow Enforcement), NIST AC-6 (Least Privilege)

Compensating: On Linux hosts, use `ss -tunap` or `netstat -tunap` to observe live outbound connections made by agent processes during a controlled skill execution in a test environment. Use `strace -e trace=network -p ` to capture all network syscalls. On Windows, use Sysmon Event ID 3 (Network Connection) filtered to the agent process name. Validate sandbox isolation by checking whether the agent process runs under a dedicated low-privilege service account (`id` on Linux, `whoami /priv` on Windows) with no write access outside its working directory.

Evidence: Capture a live `netstat -ano` (Windows) or `ss -tunap` (Linux) snapshot and the agent process tree (`tasklist /v` or `ps auxf`) BEFORE making any configuration changes. These volatile artifacts reveal which external URLs the skill runtime is actively calling and whether the agent process has elevated privileges — both directly relevant to confirming whether the post-scan substitution mechanism is reachable in your environment.

Step 3: Inventory runtime fetch behavior — for each installed skill, determine whether it makes outbound HTTP/S calls at execution time; document the external URLs or domains contacted, and verify those resources are controlled by the original publisher, not rewriteable by a third party.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlating observable behavior with known attack patterns to determine whether substitution has occurred

Controls: CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run each installed skill in an isolated test environment with Wireshark or `tcpdump -i any -w skill_runtime_capture.pcap` capturing all traffic, then filter for DNS queries and HTTP/S connections (`tshark -r skill_runtime_capture.pcap -Y 'dns || http || tls'`). For each resolved domain, perform WHOIS and passive DNS lookups (using free tools such as whois, MXToolbox, or SecurityTrails free tier) to confirm the registrant matches the declared skill publisher. Flag any domain registered after the skill's marketplace approval date as a high-priority indicator of post-scan substitution.

Evidence: The definitive forensic artifact for this threat is the delta between: (a) the external URL/domain recorded in the skill manifest at marketplace approval time and (b) the URL/domain actually contacted at runtime. Capture full PCAP of skill execution network traffic and DNS query logs (Windows DNS Client Event Log Event ID 3008, or `/var/log/syslog` DNS entries on Linux) before any network blocking is applied. If a skill reached approximately 26,000

agents as described in the AIR research, DNS resolution logs across your fleet are the fastest way to detect breadth of exposure.

Step 4: Apply continuous integrity verification — a one-time scan at submission is insufficient; implement or demand from your marketplace vendor a mechanism that re-verifies external dependency content at each execution (content hashing, signed manifests, or pinned artifact digests); align with D3-FMBV (File Magic Byte Verification) and D3-SFA (System File Analysis) principles extended to runtime-fetched content.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: implementing controls to prevent further exposure while structural remediation is developed

Controls: NIST AC-4 (Information Flow Enforcement)

Compensating: Implement a pre-execution wrapper script for each agent skill that: (1) fetches the external dependency URL, (2) computes its SHA-256 hash (`sha256sum`), (3) compares against an approved hash stored in a local pinfile, and (4) aborts execution and alerts if the hash does not match. This is achievable in ~30 lines of bash or Python and requires no SIEM. Store the approved hash pinfile in a write-protected location. For teams using osquery, write a scheduled query against the agent skill config directory to alert on manifest file modifications: `SELECT * FROM file WHERE path LIKE '/opt/agent/skills/%' AND mtime > (strftime('%s','now') - 3600);`

Evidence: Before deploying any hash-pinning or network blocking control: capture the current SHA-256 hashes of all externally fetched skill dependency artifacts in their present state (`sha256sum `), record the full HTTP response headers from those URLs (including `Last-Modified`, `ETag`, and `Content-Length`) using `curl -I`, and save a timestamped copy of each fetched artifact. These are the forensic baseline proving whether substitution has already occurred relative to the marketplace scan time.

Step 5: Restrict skill installation permissions — apply NIST AC-2 (Account Management) and CIS 6.1 (Establish an Access Granting Process) principles to skill installation: require change-control approval, limit who can add or update agent skills to dedicated administrator accounts (CIS 5.4), and log all skill install and update events per NIST AU-2 (Event Logging).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: restricting the attack surface to prevent additional malicious skills from being introduced during the active investigation

Controls: NIST AC-2 (Account Management), NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.1 (Establish an Access Granting Process), CIS 6.2 (Establish an Access Revoking Process), NIST AU-2 (Event Logging)

Compensating: On Linux-based agent hosts, enforce skill installation directory permissions so only a dedicated `agent-admin` service account has write access (`chmod 750 /opt/agent/skills/; chown agent-admin:agent-admin /opt/agent/skills/`). Enable auditd rules to log all writes to the skill directory: `auditctl -w /opt/agent/skills/ -p wa -k skill_install`. On Windows, apply NTFS ACLs to the skill directory and enable Object Access auditing (Security Event ID 4663) for write operations. These produce tamper-evident installation logs without requiring a commercial SIEM.

Evidence: Before tightening permissions: export the current ACL/permission state of the skill installation directory (`getfacl /opt/agent/skills/` or `icacls C:\AgentSkills\`), dump the list of all accounts currently holding write access, and review auth logs for any recent skill installs performed outside normal change-control windows (Linux: `grep 'skill' /var/log/auth.log`; Windows: Security Event ID 4670 — Permissions Changed). This establishes whether unauthorized skill installation has already occurred.

Step 6: Update threat model — add T1195.002 (Supply Chain Compromise) and T1102 (Web Service runtime callback) as active TTP patterns in your AI agent threat register; document the post-scan payload substitution pattern as a distinct attack scenario separate from traditional software supply chain compromise.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: lessons learned and threat model updates to improve detection for the post-scan payload substitution pattern

Compensating: Document the post-scan substitution pattern as a threat scenario in a plain-text or Markdown threat register. For each AI agent deployment, add a row: TTP = 'Post-Scan Payload Substitution via Rewritten External URL', trigger = 'skill runtime DNS query resolves to IP not matching the domain's resolution at marketplace approval time', detection method = 'DNS log delta comparison (manual or osquery)'. Share the scenario with your SOC as a Sigma rule candidate targeting DNS Client Event Log entries for skill-associated domains.

Evidence: No live-state alteration occurs in this step. Preserve the threat register update with a dated commit in version control (git) to create an auditable record. Archive the AIR research publication, Trail of Bits ClawHub disclosure materials, and any internal investigation findings as supporting evidence for the updated threat model.

Step 7: Communicate findings — brief leadership that the risk is not a patched vulnerability but an architectural gap in the marketplace model itself; frame it as a vendor assurance and procurement control issue requiring policy changes, not a one-time remediation.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicating incident findings and systemic risk to leadership and stakeholders to drive structural policy changes

Controls: NIST AC-1 (Policy And Procedures)

Compensating: Prepare a one-page executive brief (no specialist tooling required) that quantifies blast radius using the AIR research finding of approximately 26,000 agents reached by a single test skill. Include a 2-column table: left column = current one-time-scan model risk, right column = required continuous integrity verification model. Attach the skill inventory produced in Step 1 as an appendix. Distribute via standard email with read-receipt to create an auditable record of leadership notification.

Evidence: No volatile evidence capture required for this communication step. Attach as supporting documentation: the timestamped skill inventory from Step 1, the network capture evidence from Step 3, and the hash baseline from Step 4 to substantiate the risk briefing with empirical findings from your own environment.

Step 8: Monitor developments — track Trail of Bits ClawHub disclosures, Cisco AI Defense skill-scanner updates, and NVIDIA security advisories for any scanner improvements or structural mitigations; watch for CISA or NIST AI Risk Management Framework (AI RMF) guidance addressing skill marketplace integrity.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: ongoing monitoring and threat intelligence integration to detect structural mitigations or re-emergence of the post-scan substitution pattern

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Set up free RSS or email alert subscriptions for: Trail of Bits blog (trailofbits.com/blog), Cisco Security Advisories (tools.cisco.com/security/center/publicationListing.x), NVIDIA Product Security (nvidia.com/en-us/security), and CISA Known Exploited Vulnerabilities catalog. Assign one team member to review these sources weekly and log any relevant updates to the threat register maintained in Step 6. No SIEM required — a shared Markdown log with weekly entries is sufficient for a 2-person team.

Evidence: No volatile evidence capture required. Maintain a dated changelog of all vendor advisory reviews and CISA/NIST AI RMF guidance publications checked, with notes on whether any update changes the compensating controls implemented in Steps 4 and 5. This changelog serves as audit evidence that the organization is actively tracking the architectural gap rather than treating it as a closed finding.

Detection Guidance

Detection must shift from pre-deployment scanning to runtime behavioral monitoring, because the malicious payload does not exist at scan time.

Runtime egress monitoring (NIST AU-2, AU-6; CIS 8.2): Log all outbound network connections initiated by AI agent processes. Treat any skill making an HTTP/S GET or POST request to an external domain at execution

time as requiring scrutiny. Flag connections to domains not present in a pre-approved allowlist. Correlate with T1102 (Web Service) and T1071.001 (Application Layer Protocol) detections in your SIEM.

Content integrity at fetch time: Where your environment permits, implement a proxy or inline inspection layer (aligned with D3-PBWSAM, Proxy-based Web Server Access Mediation) that intercepts skill runtime fetches and computes a hash of the returned content. Compare against a baseline hash recorded at scan/approval time. A mismatch is a high-confidence indicator of post-scan payload substitution.

Skill installation and update events (NIST AU-12, AU-3): Ensure your agent orchestration platform logs every skill installation, update, and runtime dependency fetch with timestamps, source URLs, and content digests. Absence of this logging is itself a gap to remediate.

Anomaly patterns to hunt: (1) A skill that was installed weeks ago suddenly initiates outbound connections it did not make previously, possible trigger for URL rewrite. (2) A skill fetching content from a domain registered recently relative to the skill's approval date. (3) Agent processes spawning unexpected child processes after a skill fetch (T1059 execution downstream of T1102 fetch). (4) Skills impersonating known legitimate tools or services (T1036.005), verify publisher identity against the original marketplace listing.

D3FEND countermeasures to implement: D3-LAM (Local Account Monitoring) applied to the service accounts running agent processes; D3-UAP (User Account Permissions) to restrict what resources agent processes can access post-skill-load; D3-PBWSAM (Proxy-based Web Server Access Mediation) to monitor and verify skill artifact directories for unexpected modification.

Scanner disagreement as a signal: If your pipeline uses multiple skill scanners, treat high disagreement across scanners on a given skill as a risk flag, not a clean bill of health. The 0.12% consensus finding means scanner agreement is rare; scanner disagreement is normal and not reassuring.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to AIR research publication and The Hacker News article (https://thehackernews.com/2026/06/fake-ai-agent-skill-passed-security.html) for any published external callback URLs used in the demonstration`	External URL rewritten post-approval to deliver malicious payload at agent runtime; actual URL value not published in available source material	LOW
TOOL	`AI agent skill package (clean submission variant)`	Skill package leveraged via marketplace submission process to pass static analysis, then used as a loader that fetches attacker-controlled payload from a rewritten external URL at agent runtime execution	HIGH

Framework Mappings

MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter
- **T1195.002** — Compromise Software Supply Chain
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1608.001** — Upload Malware
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1027.012** — LNK Icon Smuggling
- **T1102** — Web Service
- **T1071.001** — Web Protocols

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **CM-3** — Configuration Change Control
- **AT-2** — Literacy Training and Awareness

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059	Command and Scripting Interpreter	Execution
T1195.002	Compromise Software Supply Chain	Initial-Access
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1608.001	Upload Malware	Resource-Development

Technique ID	Technique Name	Tactic
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion
T1027.012	LNK Icon Smuggling	Defense-Evasion
T1102	Web Service	Command-And-Control
T1071.001	Web Protocols	Command-And-Control

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/06/fake-ai-agent-skill-passed-securi...	T3
The sorry state of skill distribution - The Trail of Bits Blog	https://blog.trailofbits.com/2026/06/03/the-sorry-state-of-skill-di...	T3
cisco-ai-defense/skill-scanner - GitHub	https://github.com/cisco-ai-defense/skill-scanner	T3
Seven scanners for malicious AI agent skills agree on only 0.12%	https://theweatherreport.ai/posts/skill-scanner-disagreement/	T3
I Scanned Anthropic Skills and OpenClaw with Cisco's ... - YouTube	https://www.youtube.com/watch?v=k-HxQYy82Sw	T3

Generated 2026-06-24 06:18 UTC by TJS Security Command Center